Need for plans with loops

1. Action may fail to have any effect.

   EXAMPLE: Trying to break a coconut.

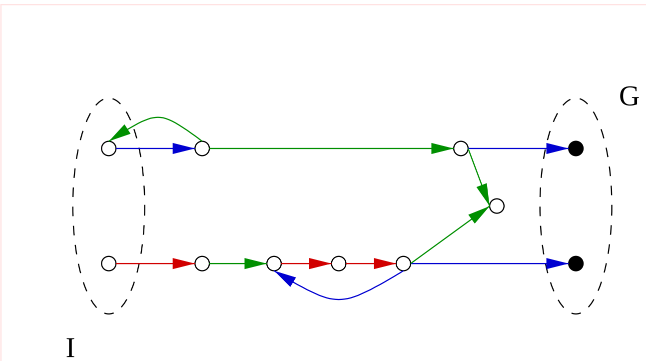
2. Action may fail and take us away from the goals.

   EXAMPLE: Building a house of cards.

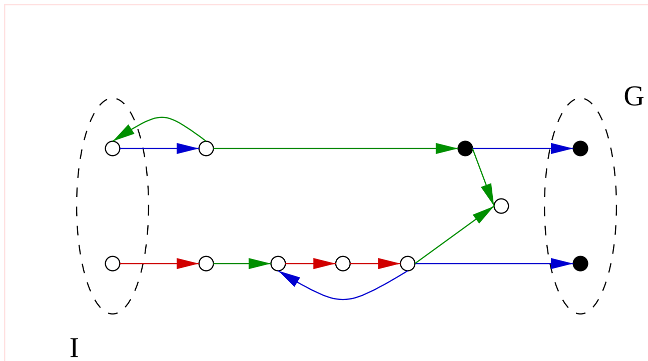
Algorithm

1. Find states with strong preimages as far as possible.
2. Find states with weak preimages until a set S of states is found so that for every state in S some action either
   (a) takes us closer to goal states, or
   (b) takes us to some state in S.
3. If initial states not covered yet, continue from 1.

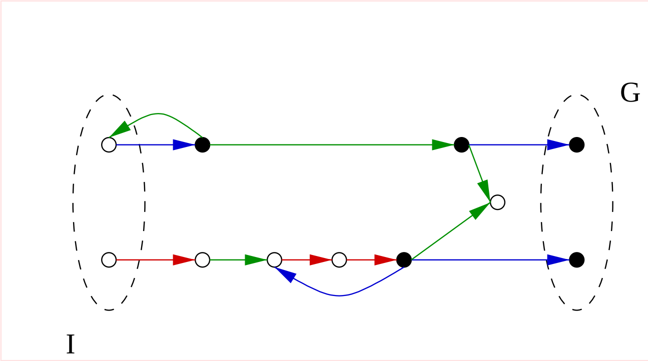
Example



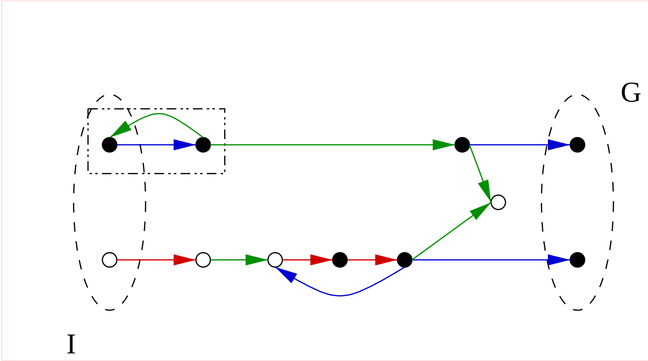
Example: strong preimages



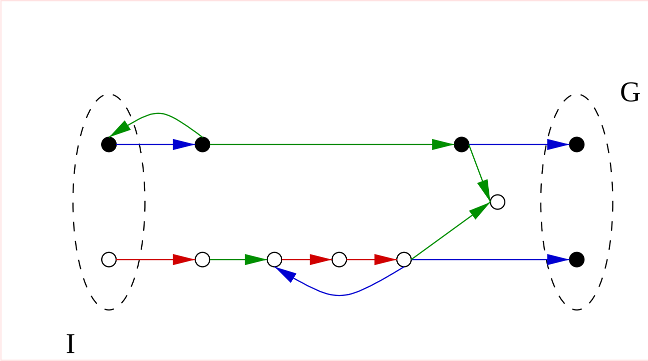
Example: weak preimages 1



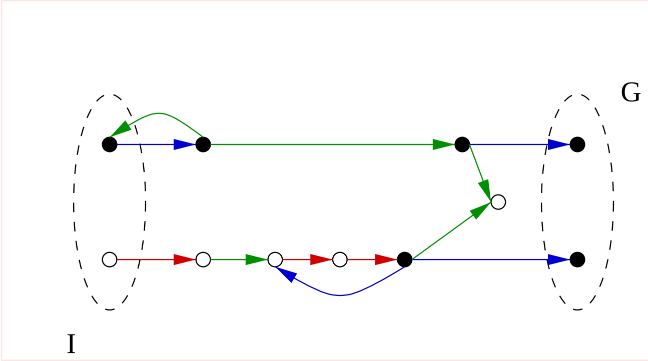
Example, weak preimages 2, loop found



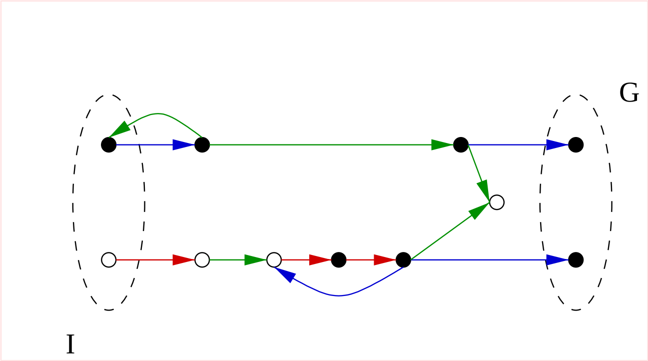
Example: continued with strong preimages



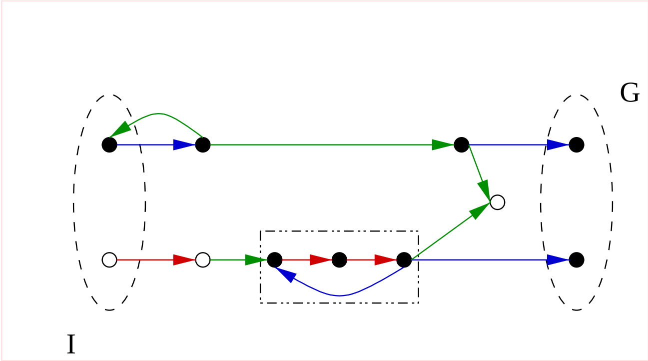
Example: weak preimages 1



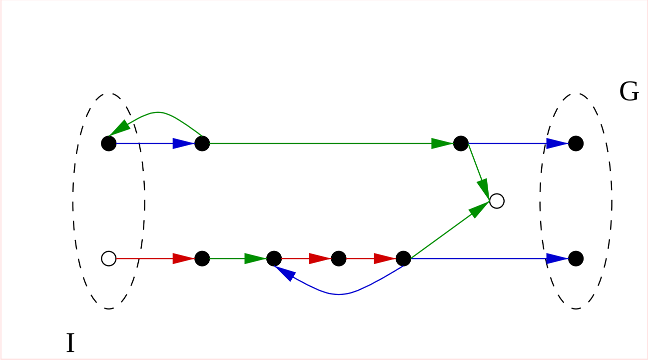
Example: weak preimages 2



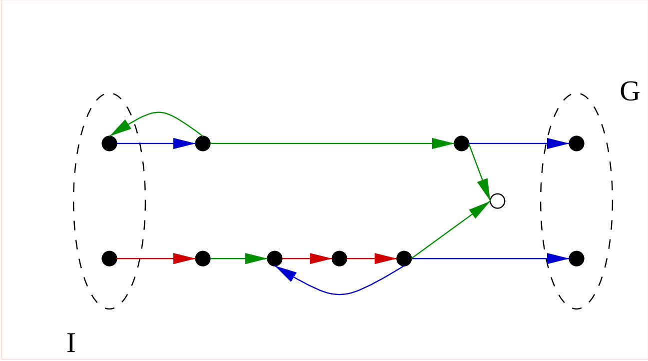
Example: weak preimages 3, loop found



Example: continued with strong preimages



Example: strong preimages, plan found



Plans with loops

Each state is assigned an operator as follows.

- States not in a loop: operator reduces distance (by at least 1).
- States in a loop: operator reduces distance (exits loop), or, the successor state is still within the loop.

Algorithm 1

PROCEDURE FOplanL(I,O,G)
D_0 := G; i := 0;
REPEAT
WHILE $I \not\subseteq D_i$ *AND* ($i = 0$ *OR* $D_{i-1} \neq D_i$) *DO*
i := $i + 1$;
D_i := $D_{i-1} \cup \bigcup_{o \in O}$ *spreimg*$_o(D_{i-1})$;
END
IF $I \not\subseteq D_i$ *THEN*
find a loop (next slide)
UNTIL $I \subseteq D_i$ *OR* $D_i = D_{i-1}$;

Algorithm 1: cont'd, detect a loop

BEGIN
i := $i - 1$;
W := D_i;
REPEAT
W' := W;
W := $W \cup \bigcup_{o \in O}$ *wpreimg*$_o(W)$;
L := prune(O,W,D_i);
UNTIL $L \neq \emptyset$ *OR* $W = W'$;
IF $L \neq \emptyset$ *THEN*
assign (weak) distances to states (next slide)
END

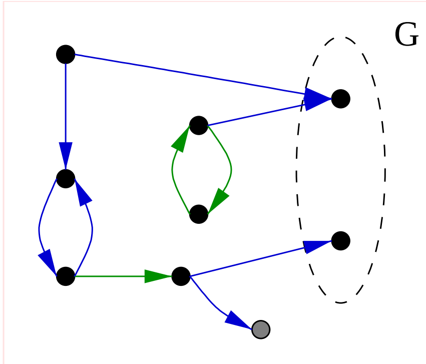
Algorithm 1: cont'd, distances for loop states

BEGIN
W := D_i;
REPEAT
W' := W;
W := $W \cup \bigcup_{o \in O}$(*wpreimg*$_o(W) \cap$ *spreimg*$_o(W \cup L)$);
i := $i + 1$;
D_i := $L \cap W$;
UNTIL $W = W'$
END

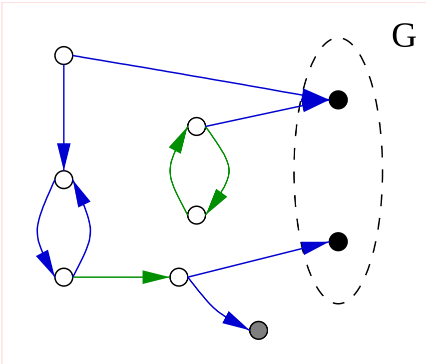
Algorithm 1: do states form a good loop?

PROCEDURE prune(O,W,G);
REPEAT
  S := G;
  REPEAT
    S' := S;
    S := $S \cup \bigcup_{o \in O}$(*wpreimg*$_o(S') \cap$ *spreimg*$_o(W)$);
  UNTIL $S = S'$;
  W' := W;
  W := $W \cap S$;
UNTIL $W = W'$;
RETURN W;

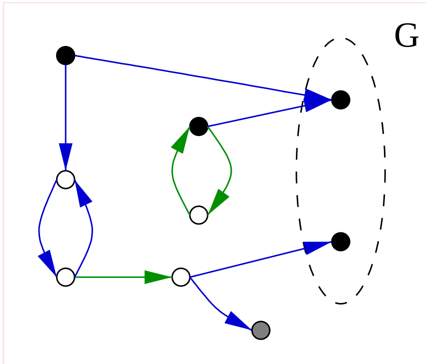
The procedure *prune*: example



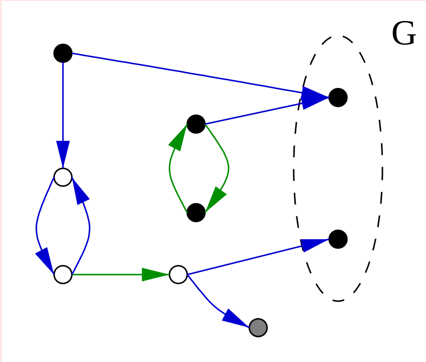
The procedure *prune*: iteration 1



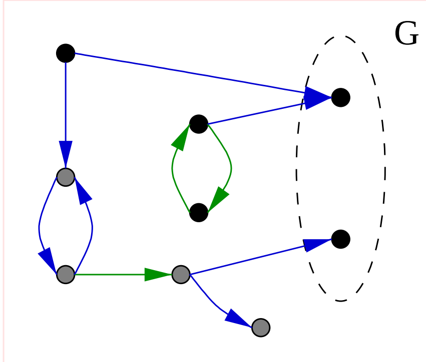
The procedure *prune*: iteration 1, 1 preimage



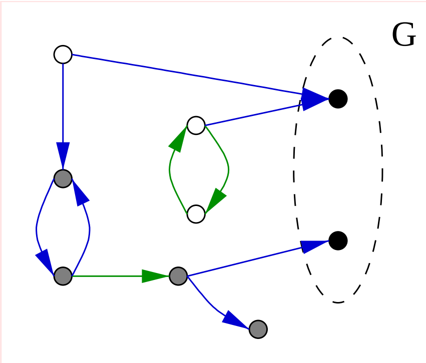
The procedure *prune*: iteration 1, 2 preimages



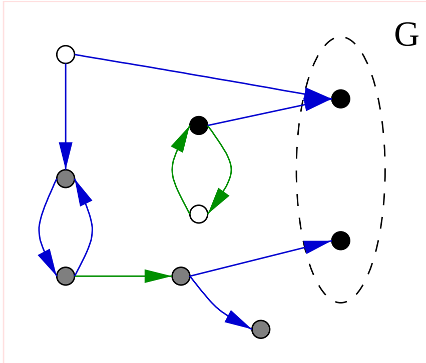
The procedure *prune*: iteration 1, bad states



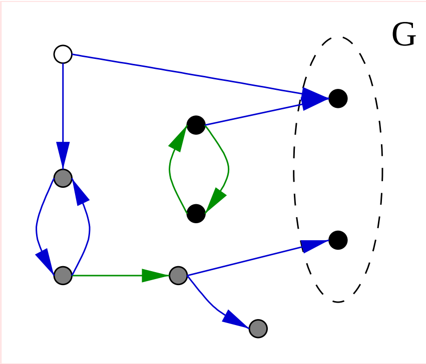
The procedure *prune*: iteration 2



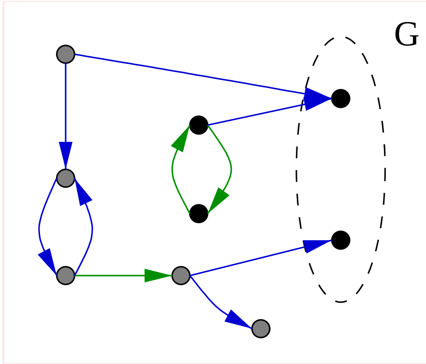
The procedure *prune*: iteration 2, 1 preimage



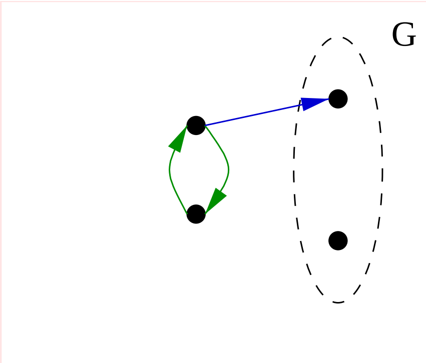
The procedure *prune*: iteration 2, 2 preimages



The procedure *prune*: iteration 2, bad states



The procedure *prune*: example, loop identified



Complexity of the planning algorithm

- The algorithm runs in polynomial time in the number of states. Because the state space may be exponential in the size of $\langle P, I, O, G\rangle$, the algorithm runs in exponential time in the size of the problem instance.
- Finding conditional plans (with full observability) is in the complexity class EXPTIME.
- Lecture notes contain a proof that testing the existence of a conditional plan (with full observability) is also EXPTIME-hard.

Algorithm 2: idea

- In fact, the subprocedure *prune* alone is sufficiently powerful for finding a plan whenever one exist.

+ Resulting algorithm is very simple.

- The algorithm does not always guarantee an upper bound for number of steps for reaching a goal state, even when one exists.

Algorithm 2

PROCEDURE FOplanL2(I,O,G)
W := G;
REPEAT
  W' := W;
  W := $W \cup \bigcup_{o \in O}$ *wpreimg*$_o(W)$;
  L := prune(O,W,D_i);
UNTIL $I \subseteq L$ *OR* $W = W'$;
IF $I \not\subseteq L$ *THEN* no plan exists;
assign (weak) distances to states (= find a plan)

Maintenance goals

- Many planning problems are not about reaching a single goal.

1. An animal: find food, eat, sleep, find food, eat, sleep, ...
2. Cleaner robot: keep the building clean.

- These problems cannot be directly formalized in terms of reachability goals: infinite (unbounded) plan execution.

Maintenance goals: formal definition

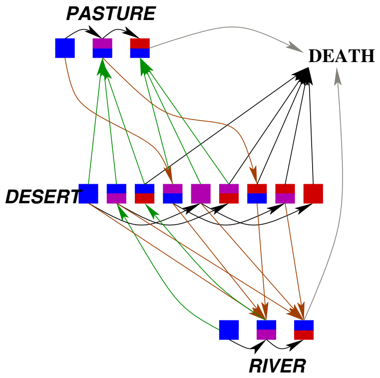
DEFINITION A plan $\pi = \langle N, b, l \rangle$ solves a problem instance $\langle P, I, O, G \rangle$ under the *Maintenance* (MG) criterion if its execution graph fulfills the following.

For all states s and s' and plan nodes $n \in N$ such that $s \models I$, if there is a path of length ≥ 0 from (s, b) to some (s', n), then $s' \models G$ and (s', n) has a successor. $\Longrightarrow$ Every state visited during plan execution is a goal state, and plan execution never ends.

Maintenance goals: an example

- The state of an animal is determined by three variables: hunger (0,1,2), thirst (0,1,2) and location (river, pasture, desert). There is also a special state called *death*.
- Thirst increases when not at river. (At river it is 0)
- Hunger increases when not on pasture. (On pasture it is 0)
- If hunger or thirst exceeds 2, the animal dies. The goal of the animal is not to die.

Maintenance goals: an example



Maintenance goals: an example

There is only one plan: go to pasture, go to desert, go to river, go to desert, ...

1. If in desert and thirst = 2 must go to river.
2. If in desert and hunger = 2 must go to pasture.
3. If on pasture and thirst = 1 must go to desert.
4. If at river and hunger = 1 must go to desert.
5. If the above rules conflict, the animal will die.

Maintenance goals: the idea of the algorithm

1. Start from goal states: they are 0-safe (the goal objective is guaranteed to hold for 0 time points *after the current state*.)
2. Given all i-safe states, compute all $i+1$-safe states: goal objective can be guaranteed to hold for $i+1$ time points.
3. $i+1$-safe states can be computed from i-safe states by the strong preimage operation.
4. For some j, j-safe states coincide with $j+1$-safe states: j-safe states are also ∞-safe.

Maintenance goals: an algorithm

PROCEDURE FOplanMAINTENANCE(I,O,G)
REPEAT
G' := G;
G := $\bigcup_{o \in O}$ (*spreimg*$_o(G) \cap G$);
UNTIL $G = G'$;
RETURN G;

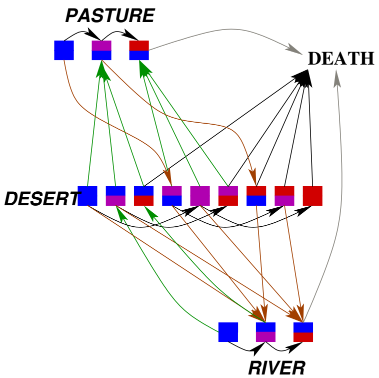
Maintenance goals: an algorithm

If FOplanmaintenance(I,O,G) returns G' such that $I \subseteq G'$, then there is a plan.

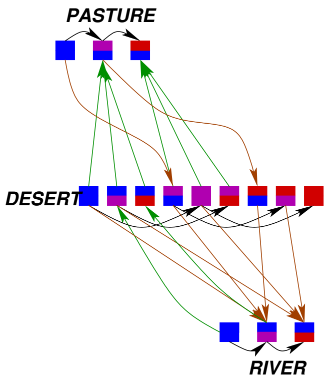
The plan:

- Any plan execution visits only states in G'.
- In state s execute an operator $o \in O$ such that *img*$_o(\{s\}) \subseteq G'$. Such an operator is guaranteed to exist for all $s \in G'$.

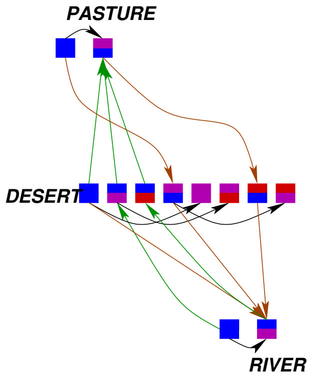
Maintenance goals: an example



Maintenance goals: an example



Maintenance goals: an example



Maintenance goals: an example